Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology model into which a plurality of enterprise data assets are mapped, the ontology model including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology model including a plurality of mapping constructs, the method comprising:

receiving [[(i)]] by a processor device in the data processing system a quantity equal to a number of ~~distinct~~ asset constructs required to represent the enterprise data assets, ~~denoted by C_{ASSET}, (ii)~~ a quantity equal to a number of ~~distinct~~ mapping constructs required to represent the mappings between the enterprise data assets and the ontology model, ~~denoted by $C_{MAPPING}$,~~ and [[(iii)]] a quantity equal to a number of ~~distinct~~ model constructs~~, denoted by C_{MODEL}~~ required to generate the ontology model from an input device;

~~evaluating~~ calculating by the processor device a metric of complexity~~, denoted by M,~~ for the enterprise information resource management system having a maximum capacity corresponding to ~~C_{ASSET}, $C_{MAPPING}$, and C_{MODEL}~~ a sum of the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$,~~

~~where f is a real-valued function of three or more real-valued parameters~~ calculated based on a function value of the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~*M* within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the enterprise information resource management system using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

2. (Previously Presented) The method of claim 1 wherein the plurality of enterprise data assets include conformed assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

3. (Previously Presented) The method of claim 2 wherein the general data schema comprises a relational database schema, the element group asset constructs comprise database tables, and the element asset constructs comprise columns of database tables.

4. (Previously Presented) The method of claim 2 wherein the general data schema comprises an XML schema, the element group asset constructs comprise XML complex types, and the element asset constructs comprise XML elements.

5. (Previously Presented) The method of claim 2 wherein the general data schema comprises a Cobol Copy Book, the element group asset constructs comprise Cobol group items, and the element asset constructs comprise Cobol elementary items.

6. (Currently Amended) The method of claim 2 wherein the ~~ontology comprises an ontology model, wherein the~~ model constructs include ontology classes and properties of the ontology classes, and wherein the model constructs further include business rules that inter-relate the properties.

7. (Previously Presented) The method of claim 6 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and further include mappings of element asset constructs into ontology properties.

8. (Currently Amended) The method of claim 6 wherein [[the]] a function $f\,(C_{ASSET}, C_{MAPPING}, C_{MODEL})$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)=f_{\mathrm{n}}(X),\ if\ C_{\mathrm{n-1}} < C < C_{\mathrm{n}},$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL,}$ and where C_0, C_1, C_2,… are cutoff points.

9. (Currently Amended) The method of claim 6 wherein [[the]] a function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)=f_{\mathrm{n}}(X),\ if\ C_{\mathrm{n-1}} < C < C_{\mathrm{n}},$$

where C is a weighted average $C = W_{ASSET}\,C_{ASSET} + W_{MAPPING}\,C_{MAPPING} + W_{MODEL}\,C_{MODEL}$ and where W_{ASSET}, $W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where C_0, C_1, C_2,… are cutoff points.

10. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system generates results for tasks, and wherein the metric [[M]] of complexity also depends on ~~a number~~ one or more of ~~distinct~~ the results generated and saved.

11. (Original) The method of claim 10 wherein the results include data transformations.

12. (Currently Amended) The method of claim 10 wherein the results include generating SQL queries for data query tasks.

13. (Currently Amended) The method of claim 10 wherein the results include generating XSLT scripts for data transformation tasks.

14. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system generates report scripts for producing reports, and wherein the metric [[M]] of complexity also depends on ~~a number~~ one or more of ~~distinct~~ the report scripts generated and saved.

15. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system records metadata, and wherein the metric [[M]] of complexity also depends on ~~a number of distinct~~ one or more metadata records recorded in the enterprise information resource management system.

16. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of a metadata repository including a plurality of metadata constructs, the method comprising:

receiving by a processor device in the data processing system a quantity equal to a number of ~~distinct~~ metadata constructs~~, denoted by C~~ from an input device;

~~evaluating~~ calculating by the processor device a metric of complexity~~, denoted by M,~~ for [[a]] the metadata repository having a maximum capacity corresponding to [[C]] the quantity of metadata constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C, X)$,~~

~~where f is a real-valued function of one or more real-valued parameters~~ calculated based on a function value of the quantity of metadata constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~M within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

17. (Previously Presented) The method of claim 16 wherein the metadata constructs comprise instances of meta-model constructs.

18. (Currently Amended) The method of claim 17 wherein the meta-data constructs comprise constructs for meta-models of asset schemas, and wherein the metric [[*M*]] of complexity also depends on ~~a number~~ one or more of the constructs for meta-models~~ constructs~~.

19. (Previously Presented) The method of claim 17 wherein at least one meta-model comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

20. (Previously Presented) The method of claim 17 wherein at least one meta-model comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

21. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, the method comprising:

receiving by a processor device in the data processing system a quantity equal to a number of ~~distinct~~ meta-model constructs~~, denoted by *C*~~ from an input device;

~~evaluating~~ calculating by the processor device a metric of complexity~~, denoted by *M*,~~ for [[a]] the metadata repository having a maximum capacity corresponding to [[*C*]] the quantity of meta-model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C, X)$,~~

~~where *f* is a real-valued function of one or more real-valued parameters~~ calculated based on a function value of the quantity of meta-model constructs and ~~*X* denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~*M* within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

22. (Previously Presented) The method of claim 21 wherein at least one meta-model comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

23. (Previously Presented) The method of claim 21 wherein at least one meta-model comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

24. (Original) The method of claim 21 wherein the meta-model constructs include descriptors for data assets.

25. (Currently Amended) The method of claim 21 wherein the ~~zero or more~~ specified additional parameters include a specified parameter [[for]] equal to a number of users of the repository.

26. (Currently Amended) The method of claim 21 wherein the ~~zero or more~~ specified additional parameters include a specified parameter [[for]] equal to a number of available features.

27. (Previously Presented) The method of claim 26 wherein an available feature comprises an ability to change a meta-model.

28. (Previously Presented) The method of claim 26 wherein an available feature comprises an ability to batch scan metadata.

29. (Currently Amended) A system for determining a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology model into which a plurality of enterprise data assets are mapped, the ontology model including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology model including a plurality of mapping constructs, the ~~method~~ system comprising:

an input device for receiving [[(i)]] a quantity of ~~distinct~~ asset constructs required to represent the enterprise data assets, ~~denoted by C_{ASSET}, (ii)~~ a quantity of ~~distinct~~ mapping constructs required to represent the mappings between the enterprise data assets and the ontology model, ~~denoted by $C_{MAPPING}$,~~ and [[(iii)]] a quantity of ~~distinct~~ model constructs~~, denoted by C_{MODEL}~~ required to generate the ontology model from a user;

a processor coupled to said input device, the processor for ~~evaluating~~ calculating a metric of complexity~~, denoted by M,~~ for the enterprise information resource management system with a maximum capacity corresponding to ~~C_{ASSET}, $C_{MAPPING}$, and C_{MODEL}~~ the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$,~~

~~where f is a real-valued function of three or more real-valued parameters~~ calculated based on a function value of the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]] inputting the calculated metric of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~receiving the metric M and for using the metric M for~~ a licensing transaction for the enterprise information resource management system using the calculated metric of complexity; and outputting the calculated metric of complexity on an output device coupled to the processor.

30. (Previously Presented) The system of claim 29 wherein the plurality of enterprise data assets include assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

31. (Previously Presented) The system of claim 30 wherein the general data schema comprises a relational database schema, the element group asset constructs comprise database tables, and the element asset constructs comprise columns of database tables.

32. (Previously Presented) The system of claim 30 wherein the general data schema comprises an XML schema, the element group asset constructs comprise XML complex types, and the element asset constructs comprise XML elements.

33. (Previously Presented) The system of claim 30 wherein the general data schema comprises a Cobol Copy Book, the element group asset constructs comprise Cobol group items, and the element asset constructs comprise Cobol elementary items.

34. (Currently Amended) The system of claim 30 wherein the ~~ontology comprises an ontology model, wherein the~~ model constructs include ontology classes and properties of the ontology classes, and wherein the model constructs further include business rules that inter-relate the properties.

35. (Previously Presented) The system of claim 34 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and further include mappings of element asset constructs into ontology properties.

36. (Currently Amended) The system of claim 34 wherein [[the]] a function $f(C_{ASSET}$ $C_{MAPPING}, C_{MODEL}, X)$, is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)=f_{n}(X), \text{ if } C_{n-1} < C < C_{n},$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL}$ and where C_0, C_1, C_2,… are cutoff points.

37. (Currently Amended) The system of claim 34 wherein [[the]] a function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)=f_{\mathrm{n}}(X), \text{ } if \ C_{\mathrm{n-1}} < C < C_{\mathrm{n}},$$

where C is a weighted average $C = W_{ASSET}\,C_{ASSET} + W_{MAPPING}\,C_{MAPPING} + W_{MODEL}\,C_{MODEL}$, and where W_{ASSET}, $W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where C_0, C_1, C_2,… are cutoff points.

38. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system generates results for tasks, and wherein the metric [[M]] of complexity also depends on ~~a number~~ one or more of ~~distinct~~ the results generated and saved.

39. (Original) The system of claim 38 wherein the results include data transformations.

40. (Currently Amended) The system of claim 38 wherein the results include generating SQL queries for data query tasks.

41. (Currently Amended) The system of claim 38 wherein the results include generating XSLT scripts for data transformation tasks.

42. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system generates report scripts for producing reports, and wherein the metric [[M]] of complexity also depends on ~~a number~~ one or more of ~~distinct~~ the report scripts generated and saved.

43. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system records metadata, and wherein the metric [[*M*]] of complexity also depends on ~~a number of distinct~~ one or more metadata records recorded in the enterprise information resource management system.

44. (Currently Amended) A system for determining a complexity of a metadata repository including a plurality of metadata constructs, the system comprising:

an input device for receiving a quantity of ~~distinct~~ metadata constructs~~, denoted by *C*~~;

a processor coupled to said input device, the processor for ~~evaluating~~ calculating a metric of complexity~~, denoted by *M*,~~ for [[a]] the metadata repository having a maximum capacity corresponding to ~~*C*, according to a formula~~

~~$M = f(C, X)$,~~

~~where *f* is a real-valued function of one or more real-valued parameters~~ the quantity of metadata constructs, wherein the metric of complexity is calculated based on a function value of the quantity of metadata constructs and ~~*X* denotes zero or more~~ specified additional parameters; [[and]] inputting the calculated metric of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~using the metric *M* for~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and outputting the calculated metric of complexity on an output device coupled to the processor.

45. (Previously Presented) The system of claim 44 wherein the plurality of metadata constructs comprise instances of meta-model constructs.

46. (Currently Amended) The system of claim 45 wherein the meta-data constructs comprise constructs for meta-models of asset schemas, and wherein the metric [[M]] of complexity also depends on ~~a number~~ one or more of the constructs for meta-models~~ constructs~~.

47. (Previously Presented) The system of claim 45 wherein at least one meta-model comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

48. (Previously Presented) The system of claim 45 wherein at least one meta-model comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

49. (Currently Amended) A system for determining a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, the system comprising:

an input device for receiving a quantity of ~~distinct~~ meta-model constructs~~, denoted by *C*~~;

a processor coupled to said input device, the processor for ~~evaluating~~ calculating a metric of complexity~~, denoted by *M*,~~ for a metadata repository having a maximum capacity corresponding to [[*C*]] the quantity of meta-model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C, X)$,~~

~~where *f* is a real-valued function of one or more real-valued parameters~~ calculated based on a function value of the quantity of meta-model constructs and ~~*X* denotes zero or more~~ specified additional parameters; [[and]] inputting the calculated metric of complexity into a transaction system, wherein the transaction system determines a price for ~~receiving the metric *M*, the transaction system further for using the metric *M* for license of~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and outputting the calculated metric of complexity on an output device coupled to the processor.

50. (Previously Presented) The system of claim 49 wherein at least one meta-model comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

51. (Previously Presented) The system of claim 49 wherein at least one meta-model comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

52. (Original) The system of claim 49 wherein the meta-model constructs include descriptors for data assets.

53. (Currently Amended) The system of claim 49 wherein the ~~zero or more~~ specified additional parameters include a specified parameter [[for]] equal to a number of users of the repository.

54. (Currently Amended) The system of claim 49 wherein the ~~zero or more~~ specified additional parameters include a specified parameter [[for]] equal to a number of available features.

55. (Previously Presented) The system of claim 54 wherein an available feature comprises an ability to change a meta-model.

56. (Previously Presented) The system of claim 54 wherein an available feature comprises an ability to batch scan metadata.

57. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology model into which a plurality of enterprise data assets are mapped, the ontology model including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology model including a plurality of mapping constructs, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining [[(i)]] by a processor in the computer a quantity of ~~distinct~~ asset constructs required to represent the enterprise data assets, ~~denoted by C_{ASSET}; (ii)~~ a quantity of ~~distinct~~

mapping constructs required to represent the mappings between the enterprise data asset and the ontology model, ~~denoted by $C_{MAPPING}$,~~ and [[(iii)]] a quantity of ~~distinct~~ model constructs~~,~~ ~~denoted by C_{MODEL}~~ required to generate the ontology model received from an input device;

~~evaluating~~ calculating by the processor a metric of complexity~~, denoted by M,~~ for the enterprise information resource management system having a maximum capacity corresponding to ~~C_{ASSET}, $C_{MAPPING}$, and C_{MODEL}~~ a sum of the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$~~

~~where f is a real-valued function of three or more real-valued parameters~~ calculated based on a function value of the quantity of asset constructs, the quantity of mapping constructs, and the quantity of model constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~M within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the enterprise information resource management system using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

58. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of a metadata repository including a plurality of metadata constructs, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining by a processor in the computer a quantity of ~~distinct~~ metadata constructs~~,~~ ~~denoted by C~~ received from an input device;

~~evaluating~~ calculating by the processor a metric of complexity~~, denoted by M,~~ for a metadata repository having a maximum capacity corresponding to [[C]] the quantity of metadata constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C, X)$,~~

~~where f is a real-valued function of one or more real-valued parameters~~ is calculated based on a function value of the quantity of metadata constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~M within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

59. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining by a processor in the computer a quantity of ~~distinct~~ meta-model constructs~~, denoted by C~~ received from an input device;

~~evaluating~~ calculating by the processor a metric of complexity~~, denoted by M,~~ for a metadata repository having a maximum capacity corresponding to [[C]] the quantity of meta-model constructs, wherein the metric of complexity is ~~evaluated according to a formula~~

~~$M = f(C, X)$,~~

~~where f is a real-valued function of one or more real-valued parameters~~ is calculated based on a function value of the quantity of meta-model constructs and ~~X denotes zero or more~~ specified additional parameters; [[and]]

~~using~~ inputting by the processor device the calculated metric ~~M within~~ of complexity into a transaction processing system, wherein the transaction processing system determines a price for ~~license of~~ a licensing transaction for the metadata repository using the calculated metric of complexity; and

outputting by the processor device the calculated metric of complexity on an output device.

60. (Currently Amended) The method of claim 1 further comprising limiting the complexity of the enterprise information resource management system to a specified limit by restricting the quantities ~~C_{ASSET}, $C_{MAPPING}$, and C_{MODEL}~~ of the asset constructs, the mapping constructs, and the model constructs.

61. (Previously Presented) The method of claim 60 wherein the specified limit is determined from a license key for the enterprise information resource management system.

62. (Previously Presented) The method of claim 60 wherein different versions of the enterprise information resource management system have different specified limits.

63. (Currently Amended) The method of claim 16 further comprising limiting the complexity of the ~~enterprise information resource management system~~ metadata repository to a specified limit by restricting the quantity [[C]] of metadata constructs.

64. (Currently Amended) The method of claim 63 wherein the specified limit is determined from a license key for the ~~enterprise information resource management system~~ metadata repository.

65. (Currently Amended) The method of claim 63 wherein different versions of the ~~enterprise information resource management system~~ metadata repository have different specified limits.

66. (Currently Amended) The method of claim 21 further comprising limiting the complexity of the ~~enterprise information resource management system~~ metadata repository to a specified limit, by restricting the quantity [[C]] of meta-model constructs.

67. (Currently Amended) The method of claim 66 wherein the specified limit is determined from a license key for the ~~enterprise information resource management system~~ metadata repository.

68. (Currently Amended) The method of claim 66 wherein different versions of the ~~enterprise information resource management system~~ metadata repository have different specified limits.

69. (Currently Amended) The system of claim 29 further comprising a controller restricting the quantities ~~C_{ASSET}, $C_{MAPPING}$, and C_{MODEL}~~ of the asset constructs, the mapping constructs, and the model constructs such that the complexity of the enterprise information resource management system does not exceed a specified limit.

70. (Previously Presented) The system of claim 69 wherein the specified limit is determined from a license key for the enterprise information resource management system.

71. (Previously Presented) The system of claim 69 wherein different versions of the enterprise information resource management system have different specified limits.

72. (Currently Amended) The system of claim 44 further comprising a controller restricting the ~~quantities C~~ quantity of metadata constructs such that the complexity of the ~~enterprise information resource management system~~ metadata repository does not exceed a specified limit.

73. (Currently Amended) The system of claim 72 wherein the specified limit is determined from a license key for the ~~enterprise information resource management system~~ metadata repository.

74. (Currently Amended) The system of claim 72 wherein different versions of the ~~enterprise information resource management system~~ metadata repository have different specified limits.

75. (Currently Amended) The system of claim 49 further comprising a controller restricting the ~~quantities C~~ quantity of meta-model constructs such that the complexity of the ~~enterprise information resource management system~~ metadata repository does not exceed a specified limit.

76. (Currently Amended) The system of claim 75 wherein the specified limit is determined from a license key for the ~~enterprise information resource management system~~ metadata repository.

77. (Currently Amended) The system of claim 75 wherein different versions of the ~~enterprise information resource management system~~ metadata repository have different specified limits.